

jc715 U.S. PTO
02/22/00

Please type a plus sign (+) inside this box [+]

PTO/SB/05 (12/97)

Approved for use through 09/30/00. OMB 0651-0032

Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL
(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No. 003388.P001C

Total Pages 6

First Named Inventor or Application Identifier Paul Carl Kocher

Express Mail Label No. EL371008249US

jc525 U.S. PTO
09/511645

02/22/00

**ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, D. C. 20231**

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. X Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)
2. X Specification (Total Pages 43)
(preferred arrangement set forth below)
- Descriptive Title of the Invention
 - Cross References to Related Applications
 - Statement Regarding Fed sponsored R & D
 - Reference to Microfiche Appendix
 - Background of the Invention
 - Brief Summary of the Invention
 - Brief Description of the Drawings (if filed)
 - Detailed Description
 - Claims
 - Abstract of the Disclosure
3. X Drawings(s) (35 USC 113) (Total Sheets 15)
4. X Oath or Declaration (Total Pages 2)
- a. Newly Executed (Original or Copy)
 - b. X Copy from a Prior Application (37 CFR 1.63(d))
(for Continuation/Divisional with Box 17 completed) (**Note Box 5 below**)
 - i. DELETIONS OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. X Incorporation By Reference (useable if Box 4b is checked)
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. Microfiche Computer Program (Appendix)

SCANNED 7

7. ☐ Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
a. ☐ Computer Readable Copy
b. ☐ Paper Copy (identical to computer copy)
c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. ☐ Assignment Papers (cover sheet & documents(s))
9. ☐ a. 37 CFR 3.73(b) Statement (where there is an assignee)
☐ b. Power of Attorney
10. ☐ English Translation Document (if applicable)
11. ☐ a. Information Disclosure Statement (IDS)/PTO-1449
☐ b. Copies of IDS Citations
12. ☒ Preliminary Amendment
13. ☒ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
14. ☐ a. Small Entity Statement(s)
☐ b. Statement filed in prior application, Status still proper and desired
15. ☐ Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. ☐ Other: _____

17. If a **CONTINUING APPLICATION**, check appropriate box and supply the requisite information:
☒ Continuation ☐ Divisional ☐ Continuation-in-part (CIP)
of prior application No: 09/103,656

18. Correspondence Address

☐ Customer Number or Bar Code Label _____
(Insert Customer No. or Attach Bar Code Label here)
or

☒ Correspondence Address Below

NAME Daniel M. De Vos
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

ADDRESS 12400 Wilshire Boulevard
Seventh Floor

CITY Los Angeles STATE California ZIP CODE 90025-1026

Country U.S.A. TELEPHONE (408) 720-8598 FAX (408) 720-9397

12/01/97

- 2 -

PTO/SB/05 (12/97)

Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

SCANNED

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Paul Carl Kocher

Serial No. Not Yet Assigned

Filed:

For: Apparatus and Method for
Demonstrating and Confirming the
Status of Digital Certificates and
Other Data

Examiner: Not yet assigned

Art Unit: Not yet assigned

Which is a Continuation of:

Serial No.: 09/103,656

Filed: June 23, 1998

Which is a Divisional of:

Patent No.: 5,903,651

Filed: May 14, 1996

Assistant Commissioner for Patents
Washington, D.C. 20231

PRELIMINARY AMENDMENT

Sir:

The Applicant respectfully requests the Examiner to enter the following
amendment and to consider the following remark:

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: EL371008249US Date of Deposit: February 22, 2000
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service
"Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has
been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Name: Jane WolfeSignature: Jane WolfeDate: February 22, 2000

In the Specification:

On page 7, line 10, please replace "There are thus" with --Thus, there are--.

On page 11, lines 3-4, please replace "reduce greatly" with --greatly reduce--.

On page 15, lines 1-2, please replace "leaf node" with --range--.

On page 19, line 5, please replace "item is" with --items are--.

On page 23, line 13, please delete "be".

On page 28, line 3, please replace "the leaf node whose range spans" with --the range that spans".

On page 28, line 5, please replace "intermediate" with --additional--.

On page 28, line 15, please replace "want verify" with --want to verify--.

Please insert as the first sentence of the specification -- This is a continuation of application Serial No. 09/103,656, filed June 23, 1998, pending, which is a divisional of Patent No. 5,903,651, filed on May 14, 1996.--

REMARK

Specification Corrections

Applicant respectfully submits amendments to certain parts of the specification to make them more consistent with the remainder of the specification and to improve readability.

Invitation for a telephone interview

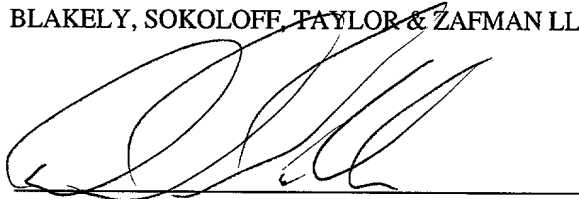
The Examiner is invited to call the undersigned at 408-720-8598 if there remains any issue with allowance of this case.

Charge our Deposit Account

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP



Date: 2/22, 2000

Daniel M. De Vos
Reg. No. 37,813

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025-1026
(408) 720-8598

003388.P001C

UNITED STATES PATENT APPLICATION

FOR

**APPARATUS AND METHOD FOR DEMONSTRATING AND CONFIRMING
THE STATUS OF A DIGITAL CERTIFICATES AND OTHER DATA**

Inventor:

Paul Carl Kocher

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026
(408) 720-8598

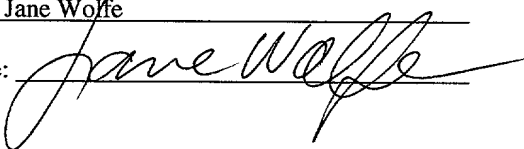
File No.: 003388.P001C

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: EL371008249US Date of Deposit: February 22, 2000

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service
"Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has
been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Name: Jane Wolfe

Signature: 

Date: February 22, 2000

APPARATUS AND METHOD FOR DEMONSTRATING AND CONFIRMING
THE STATUS OF DIGITAL CERTIFICATES AND OTHER DATA

References cited:

5

U.S. PATENT DOCUMENTS

4,405,829	9/1983	Rivest et al.	178/22.1
4,309,569	1/1982	Merkle	178/22.08
4,881,264	11/1989	Merkle	380/25

PUBLICATIONS

ISO 9594-8, *Information Technology - Open Systems Interconnection*
- *The Direction: Authentication framework*, 15 September 1995.

National Institute of Standards and Technology, NIST FIPS PUB
190, "Secure Hash Standard," U.S. Department of Commerce, May
1993.

National Institute of Standards and Technology, NIST FIPS PUB
186, "Digital Signature Standard," U.S. Department of Commerce,
May 1994.

THE
NEW
YORK
PUBLIC
LIBRARY
ASTEN LENOX TILDEN FOUNDATION
100 N. Y.

[illegible]

5 systems for constructing digitally-signed lists and determining whether particular values are present on such lists. The invention has specific application to revocation of digital certificates or other types of digital data items and for determining whether such data items have been revoked.

DESCRIPTION OF THE BACKGROUND ART

Asymmetric (public key) cryptologic techniques are widely used in security protocols for applications such as secure e-mail, electronic commerce, encrypted voice and video communications, and security on the World Wide Web. The RSA cryptosystem, described in U.S. patent 4,405,829 to Rivest et al (1983), and the Digital Signature Algorithm (DSA), described in U.S. patent 5,231,668, to Kravitz, are examples of asymmetric functions. Asymmetric cryptosystems typically involve a secret private key, which is used for digital signing or decryption, and a non-confidential public key derived from the private key, which is used for signature verification or encryption. For general information about RSA, DSA, and other asymmetric cryptosystems, the reader is referred to *Applied Cryptography*.

Before using a public key to encrypt a confidential message or verify a signature, a party in an electronic communications protocol generally must confirm the identity of the party holding the private key. An electronic communications protocol defines conventions allowing two or more computers or other electronic devices to exchange digital data or messages via

a communications channel. Without such confirmation, an attacker could substitute a legitimate public key with another for which the attacker knows the private key. Digital certificates are the most common solution to this problem. The holder of the private key provides its corresponding public key to a widely-trusted Certificate Authority (CA) along with acceptable identification. The CA then issues a certificate, which typically consists of a digital signature on a specially-formatted block of data containing the user's name, the user's public key, the certificate issuance and expiration dates, and the certificate serial number. The recipient of a digital certificate who trusts the issuing CA can use the CA's (already trusted) public key to verify the signature. If the signature is valid and if the CA is trustworthy, the recipient can trust that the person identified in the certificate holds the private key corresponding to the public key in the certificate. The ISO 9594-8 standard defines techniques and data formats for computing and verifying digital signatures and certificates.

Certificates often need to be revoked due to unexpected events such as compromise, theft, or loss of the device containing the private key. A certificate might also need to be revoked if a user has lost the privileges granted by the

certificate. In general, a certificate's status might be good, revoked, or pending, as well as other possibilities that will be appreciated by those skilled in the art.

In large open networks such as the Internet, certificate status determination, specifically certificate revocation, presents enormous challenges. The Internet is expected to have hundreds of millions of users worldwide soon. It is desirable that certificate revocation messages propagate as quickly as possible to all users who might otherwise accept the invalid certificate. There are thus difficult design constraints which a successful system must satisfy:

1. Network applications are sensitive to latency. A good solution should minimize the number of additional network connections and data exchanges required.

2. The system must work on a global scale and work on a broad range of systems with different levels of connectivity.

3. The system must be distributable so that critical information can be cached in many locations at once to minimize the number of long-distance network connections.

4. The system must be cryptographically secure.

Previous certificate revocation mechanisms, such as ISO 9594-8, use a type of digitally-signed structure called a Certificate Revocation List (CRL) which is issued periodically by the CA and lists the serial numbers of certificates issued by the CA which have been revoked. FIG. 1 shows the structure of a typical CRL 101, which consists of the issuer's name, a field identifying the signature algorithm, the date and time of issuance, and a list of revoked certificates, followed by a digital signature of the above information. To determine if a particular certificate is revoked, one obtains an up-to-date CRL from the appropriate CA, verifies that the digital signature in the CRL is valid, then searches the list of revoked certificates to determine whether the certificate in question is revoked. If the certificate is not on the list, it is assumed to be valid.

Because the complete CRL must be obtained and verified to determine the revocation status of a single certificate, CRLs do not scale well to large networks. In particular, existing certificate revocation mechanisms suffer from a number of disadvantages:

(a) CRLs can become extremely large, making them inefficient to transmit or process. For example, a very large system might have several million revoked certificates, resulting

in CRLs which are many megabytes in size. To determine whether a particular certificate is valid, one must download a recent CRL in its entirety and process the entire list to verify the digital signature. For a large network, the required network bandwidth can be prohibitively large, especially if every user needs to download new CRLs often. The time required to process a large list can also be an issue.

(b) Only mechanisms recognized and supported by the certificate recipient can be used to revoke certificates. In most cases, only revocations issued by the CA are accepted. Additional revocation mechanisms cannot be added easily.

(c) Because CAs are entrusted with both certificate issuance and revocation, physical destruction of a CA's private key could result in a situation where certificates could no longer be revoked without revoking the CA's public key.

(d) Verifiers must be able to obtain up-to-date CRLs from every supported CA. Certificate chaining makes this particularly difficult since there can easily be an extremely large number of CAs and multiple CAs per certificate chain.

SUMMARY OF THE INVENTION

Accordingly, it is an object of the invention to reduce greatly the processing effort, network bandwidth, network
5 latency, data storage, and data replication requirements needed to determine whether a particular certificate has been revoked. In particular, the invention allows certificate status to be determined without knowledge of the entire list of revoked certificates and without having to search the entire list of
10 revoked certificates.

Another object of the invention is to simplify the addition of new revocation mechanisms, such as revocation by certificate holders, without altering existing revocation
15 mechanisms.

Another object of the invention is to allow revocations from many CAs to be included efficiently in a single database, thereby allowing a single trusted source for certificate
20 revocation messages.

Another object of the invention is to provide a
20 certificate revocation system whose operation is open to public

scrutiny to ensure that certificates are not maliciously revoked and that revoked certificates are not marked as valid.

In general, the invention can be used to determine whether data items of any type are present on a digitally-signed list without requiring that the verifier retrieve the entire list. It should be readily apparent to a reader skilled in the art that the problem of determining securely whether a data item belongs to a list of data items has applications to many problems beyond certificate revocation. For example, an Internet user might want to determine whether a digitally-signed Java application has been revoked as having harmful side effects.

Briefly, the present invention typically includes at least one tree issuing device, one or more confirmation issuers, and at least one verification device.

The tree issuing device assembles a list of data items, which can have any content but would typically be a list of serial numbers identifying revoked digital certificates. The issuer sorts the list, optionally removes any duplicate entries, then adds a beginning-of-list marker and an end-of-list marker. Each pair of adjacent entries in this sorted list specifies a range between which there are no list entries. Except for the

beginning and end markers, each list entry appears in two ranges, once as a minimum value and once as a maximum value. A hash tree is then constructed where leaf nodes correspond to ranges in the list. Because the tree's leaf nodes define intervals, this

5 structure is referred to as an interval hash tree. A binary tree such as those described in U.S. patent 4,309,569 to Merkle (1982) would typically be used, but those skilled in the art will appreciate that a variety of other hash tree structures are also suitable (for example, as described in U.S. Patent 4,881,264 to Merkle (1989)). Merkle uses hash trees to reduce the cost per signature when computing a large number of digital signatures by combining a large number of items into a single root node which can be digitally signed. Merkle's hash tree techniques produce assurances that particular items have been digitally signed.

10 However, Merkle's hash trees do not provide the capability disclosed herein, of cryptographically demonstrating that particular items were not included in the tree (except in the highly inefficient case where the verifier obtains the entire tree and searches it for the particular candidate item).

20 The tree issuing device digitally signs the tree's root node (or nodes, if the chosen tree structure has multiple roots)

with other data which would typically identify the issuer's identity and the date and time.

A confirmation issuer obtains the hash tree including the root node digital signature. The tree may be obtained either separately from, or together with, its digitally signed root. In the latter case, the hash tree shall be considered to include its digitally signed root(s). These values could be obtained directly from the tree issuer, be computed independently, or be obtained from other sources. The confirmation issuer can be the same device as the tree issuer, or could be an independent device connected to the tree issuer via a communications channel. Confirmation issuers might also be included in devices including, but not limited to, network servers, firewalls, or directory servers and might be assigned to serve a particular region of a network or subnetwork.

The verification device (verifier) begins with a "candidate data item" whose status on the list is to be determined. The verifier sends the candidate data item (or a representation thereof) to the confirmation issuer. The confirmation issuer locates a leaf node whose minimum range value is no larger than the candidate data item and whose maximum range value is no smaller than the candidate data item. The

confirmation issuer then sends the verifier the appropriate leaf node, the digitally-signed root node, and the additional nodes needed to cryptographically derive the root node from the leaf node. By cryptographically determining that the particular leaf
5 can be used to reconstruct the root node, the verifier gains cryptographic assurance that the leaf was part of the original tree whose root node was digitally signed. The leaf is said to be cryptographically bound to the root. Note that the confirmation issuer does not need a private key, or knowledge of tree issuer's private key, since it does not generate any new digital signatures.

The verifier confirms that the signature on the header is correct and comes from a trusted tree issuer; that the date, name, and other information included in the root node digital
signature are appropriate; that the root node can be constructed from the leaf node using the specified supporting nodes; and that the candidate data item is within the range specified by the given leaf node. If any of the above verification steps fail, the assurance is bad and item's status on the list cannot be
20 determined. If the verification steps are successful and either range endpoint equals the data item, the verifier has cryptographic assurance that the data item is present on the

list. If the smaller range endpoint is less than the data item and the larger endpoint is larger than the item, the verifier has cryptographic assurance that the item is not present on the list.

In addition to the embodiments described above, another
5 illustrated embodiment does not use ranges, but rather uses a
hash tree constructed from a sorted list of data items such that
two adjacent leaves spanning a candidate data item provide
cryptographic assurance that the candidate data item is not
present on the list. Yet another illustrated embodiment does not
10 use ranges, but rather uses the hash tree to build digitally
signed assertions that specific items are not on the list. Still
another illustrated embodiment does not use hash trees, but
rather uses individually signed ranges.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows the contents of a traditional certificate revocation list which does not use the present invention.

5 FIG. 2 shows an exemplary technique for preprocessing data items.

FIG. 3 show a particular preprocessing technique appropriate for digital certificates.

FIG. 4 shows the construction of a set of ranges from a list of data items.

FIG. 5 shows a set of ranges made from a sorted list of data items.

FIG. 6 shows the structure of a binary interval hash tree made from a list containing six ranges.

15 FIG. 7 is a flowchart showing steps to construct an interval hash tree from a set of ranges.

FIG. 8 shows a degenerate interval hash tree.

FIG. 9 shows the contents of a digitally-signed root node.

FIG. 10 shows an alternate embodiment of the invention, without a hash tree, in which the individual ranges are signed directly.

FIG. 11 shows the steps taken by the confirmation issuer to issue a confirmation for a specified candidate item.

FIG. 12 is a flowchart showing steps to construct a list of supporting nodes cryptographically binding a leaf node to the root node.

FIG. 13 shows the steps typically taken by the verifier to determine whether a confirmation message provides acceptable cryptographic proof of the status of an item.

FIG. 14 is a flowchart showing how to use a set of supporting nodes to verify the cryptographic binding between a leaf node and the root node.

FIG. 15 outlines the operation of a communications system which a certificate holder uses to obtain confirmation messages which it provides to certificate acceptors as cryptographically-secure evidence that the certificate has not been revoked.

DETAILED DESCRIPTION OF THE INVENTION

Methods and apparatuses are disclosed for constructing efficient cryptographically secure assertions as to whether
5 candidate item is present on a list. In one embodiment of the invention, data items on the list are converted into a set of ranges having data items as endpoints such that there are no data items on the list between the endpoints of any range. The ranges are used as leaf nodes to construct a hash tree, then the tree's root node is digitally signed by the tree issuer. A verifier can determine the validity of a leaf node by checking the path from the leaf to the root node and by checking the digital signature on the root node. A valid leaf node with a range endpoint equal to a candidate data item provides cryptographic assurance that the candidate data item is present on the list. A valid leaf node with one range endpoint larger than the candidate data item and one range endpoint smaller than the candidate data item provides cryptographic assurance that the candidate data item is not on the list.

Hash Tree Construction and Issuance

For certain kinds of data items, preprocessing using a collision-free function (CFF) may be performed before the data items are used for tree construction. A CFF is a function for which it is believed to be computationally unfeasible to find two different messages X and Y such that $CFF(X) = CFF(Y)$. The identity function $I(X) = X$ is a CFF and may be used, although for larger data items a cryptographic hash function is generally more efficient. Cryptographic hash functions include SHA or MD5 and are used to reduce the size of messages (i.e. the size of $H(X)$ is less than the size of X) yet $H(X)$ is cryptographically collision-free. The exemplary preprocessing technique shown in FIG. 2 uses a cryptographic hash function to reduce the size of the items. Each data item 201 is hashed at step 202 to produce a fixed-length processed item 203. For example, if the data item consisted of a CA name ("Sample CA") followed by a four-byte serial number (decimal 123456789 with hexadecimal representation 07 5B CD 15), the processed item might be $SHA("Sample CA" \parallel 07 \ 5B \ CD \ 15)$, where " \parallel " denotes concatenation, or:

9D 76 7D 83 1D 85 A2 A8 35 95 08 DB 91 F2 AA DC D8 DD

04 AD.

Data items of specific types may use different kinds of preprocessing. FIG. 3 illustrates a particular preprocessing technique appropriate for data items such as digital certificates. The certificate issuer name 301 is hashed at step 303, and at step 304, the hashed issuer name is concatenated with the certificate serial number 302 to produce the processed digital certificate 305. (The serial number could also be hashed before concatenation.) For example, a certificate with a 32-bit (4-byte) serial number 123456789 whose CA name is "Sample CA" would have a 24-byte list entry consisting of SHA ("Sample CA") followed by the byte representation of 123456789. In particular, the hexadecimal representation would be:

E2 CA 64 56 40 BE 99 AC CA 9D 3A 9B 02 97 0D 1E F2 95
8E A0 07 5B CD 15

FIG. 4 shows one way to use a computer to convert the set of preprocessed data items into a set of ranges. At step 401, the data items are assembled into a list stored in a computer-readable memory, which is then sorted in ascending order at step 402. At step 403, markers are added to denote the beginning and end of the list. If the data items consist of a 160-bit SHA output, the beginning-of-list marker might consist of 160 zero bits, and the end-of-list marker might consist of 160

one bits. Every pair of adjacent list entries then defines a range where the list entries are the range endpoints. There are no entries on the list which lie between these endpoints (if a value between the endpoints was present on the list, the sorting operation would have placed it between the range endpoints and the range endpoints would thus no longer be adjacent on the sorted list). At step 404, a data structure is constructed specifying the range, typically encoding the range as the minimum (the lesser of the two list entries) and the range maximum (the greater list entry). Other formats for the range data structure are also possible, such as a range endpoint and length, a range midpoint and length, etc. Ranges can also be broken into subranges, although in this case range endpoints would not necessarily correspond to data items on the list. In some cases it is helpful to add additional markers in places other than the beginning and end of the list. For example, if the digital certificate preprocessing technique of FIG. 3 is used and certificates from multiple certificate issuers are present on the list, additional markers might be placed at the beginning and end of the list region corresponding to each certificate issuer. Ranges would only be issued within regions belonging to supported CAs. In general, additional start and stop signals are helpful

to define ranges excluded from the list or which are otherwise noteworthy.

If the initial data set contains n data items, there will be $n+2$ entries in the sorted list due to the addition of the beginning and end markers. There are $n+1$ possible pairs of adjacent entries in the sorted list, so there will be $n+1$ range structures. FIG. 5 shows a set of ranges constructed from a sorted list 501 of five data items $I_0..I_4$. The first range 502 goes from the beginning-of-list marker to the first data item, I_0 . The next range 503 goes from I_0 to I_1 . Subsequent ranges 504 are I_1-I_2 , I_2-I_3 , and I_3-I_4 . The final range 505 is I_4 through the end-of-list marker.

A hash tree is then be built from the sorted list of ranges. A hash tree is a hierarchical data structure comprising a plurality of leaf nodes combined using a cryptographic function to form a lesser number of root nodes such that using one or more applications of the cryptographic hash function it is possible to cryptographically transform any leaf node into a root node. A hash tree where the leaves are intervals (ranges) is called an interval hash tree.

can be done. Step 708 also replaces n with the number of nodes in the new level by dividing by 2 while rounding upward. At step 709 the device then determines whether more than one node remains. If so, it returns to step 702. Otherwise the device
5 finishes at step 710, returning the root node which is node $N_{x,0}$.

Although the preferred embodiment uses binary trees (see, e.g., U.S. patent 4,309,569 to Merkle (1982)), other tree structures are also possible. Variant tree structures, such as those having more than one root node, which combine more than two nodes at once, or which otherwise deviate from the binary tree are called degenerate trees. In some situations it may be desirable to have multiple roots since this can shorten the paths from leaves to roots. FIG. 8 shows an example of a degenerate tree in which groups of three nodes 801 (instead of two) are combined, a level x node 802 (i.e., $N_{1,1}$) is used in computing more than one level $x+1$ nodes 802 (e.g., $N_{2,0}$ and $N_{2,1}$), and there are two root nodes 803 and 804. It will be apparent to one skilled in the art that a wide variety of degenerate tree structures can be constructed. For example, U.S. patent
20 4,881,264 to Merkle (1989) describes several degenerate hash tree structures which may be used in connection with the present invention.

Referring to FIG. 9, after constructing the hash tree, the tree issuer uses RSA, DSA, or another signature algorithm to digitally sign the tree's root node 901, the date and time of the tree's issuance 902, the date and time of the next issuance (optional) 903, and the total number of nodes in the tree 904. The structure might also include other information, for example (but not limited to) the signing algorithm identifier, the tree issuer's name, and the root node of the previous tree.

To summarize, the tree issuer thus performs the following steps:

1. Construct the list of items,
2. Convert list into a set of ranges,
3. Build an interval hash tree from the ranges,
4. Digitally sign the hash tree's root node, and
5. Publish the hash tree and signed root node.

The foregoing illustrates the preferred embodiment of the invention in which hash trees are used. Alternatively, FIG. 10 shows a treeless variant in which the individual ranges 1001 are signed directly at 1002 to produce a set of signed ranges 1003.

Confirmation Issuance

Corresponding to the embodiments of the invention disclosed in FIGS. 2-9, FIG. 11 outlines a process for issuing cryptographic assurances as to whether specific items are on a list. The list is the plurality of data items used to generate the ranges for an interval hash tree, and the confirmation will directly demonstrate whether a candidate data item belongs to the plurality of items (i.e., is present on the list). At step 1101, the confirmation issuer first obtains the interval hash tree (or information allowing the confirmation issuer to construct at least the required portions of the tree) along with its digitally signed root node. At step 1102, the confirmation issuer receives a candidate data item for which the confirmation is to be constructed. At step 1103, the confirmation issuer performs any required preprocessing steps. At step 1104, the confirmation issuer identifies a leaf node representative of the candidate item. In one embodiment of the invention, the identified leaf node has a lesser range endpoint which is no larger than the candidate item and a larger range endpoint which is no smaller than the candidate item. Next, at step 1105, a list is made specifying the intermediate nodes needed to cryptographically reconstruct the path binding the leaf to the root node (FIG. 12

describes in detail the steps required to locate the appropriate intermediate nodes). At step 1106, the final confirmation including the contents of the leaf node whose range spans the candidate data item, the number of the specified leaf node in the tree, the intermediate nodes which cryptographically bind the leaf to the root node, and the digitally-signed root node, is produced. Note that the confirmation does not include the entire list of data items represented by the leaf nodes of the interval hash tree. Finally, at step 1107, the confirmation issuer issues the confirmation to the party requesting it. The requesting party may be either a party wishing to know the status of the candidate item, or it may be an intermediate in a communications channel. For example, the requesting party might be the owner of a certificate which will store the confirmation and supply it to any other parties which want verify the status of its certificate. In this case, the confirmation is communicated by the confirmation issuer to the verifier via the certificate holder. Alternatively, the confirmation might be requested by the party which wishes to verify the certificate's status.

FIG. 12 describes steps to find the intermediate nodes needed to cryptographically bind a leaf node ($N_{0,i}$) to the root node of a binary interval hash tree. The process begins at step

1201 by setting node counter n to the total number of leaf nodes in the tree and i to the vertical position of the leaf node to be bound to the root. At step 1202, x is initialized to zero. At step 1203, the device computes $i \oplus 1$, where " \oplus " denotes an exclusive-OR operation. Equivalently, step 1203 could be defined as $y = i + 1 - 2(i \bmod 2)$. At step 1204, if y is less than the total number of level x nodes (hashes) then $N_{x,y}$ is added to the list of hashes binding the specified leaf to the root. At step 1205, i is divided by 2 and rounded upward to find the vertical position of the level $x+1$ node leading to the root. At step 1206, n is updated to equal the number of level $x+1$ nodes. At step 1207, x is incremented. At step 1208, the device tests whether n is larger than 1 and, if so, loops back to step 1203.

To summarize, the confirmation issuer obtains the hash tree and signed root, receives a confirmation request, and constructs and issues the confirmation. When the hash tree is about to expire or a new tree is available, the confirmation issuer obtains an updated tree.

Verification

FIG. 13 outlines the steps taken by a verifier to use a confirmation message stored in a computer readable memory to

cryptographically determine the status of a candidate data item with respect to the list. At step 1301, the verifier first uses the tree issuer's public key (which is assumed to be previously known and trusted) to check the digital-signature on the root node. At step 1302, the verifier then checks the date and time of the tree's issuance, along with any other auxiliary information signed with the root, to ensure that the tree is acceptable. At step 1303, the verifier confirms that the leaf node is representative of the data item. In particular, the leaf's lesser range endpoint should be no larger than the data item and the larger range endpoint should be no smaller than the data item. At step 1304, the verifier uses the supporting nodes to check the cryptographic binding between the leaf node and the root node (FIG. 14 shows the steps required to verify the cryptographically binding between a leaf node and the root node in a binary tree). If any steps fail, the verifier skips to step 1308 and the confirmation is invalid. If all steps above are successful, the verifier checks at step 1305 whether either endpoint of the leaf node's range equals the candidate data item. If the candidate data item lies between the endpoints of the range, the verifier concludes at step 1306 with assurance that the item is not on the list. If the item in question equals one

of the range endpoints, the verifier concludes at step 1307 with assurance that the item is on the list.

FIG. 14 outlines the process of using a set of supporting nodes to verify the cryptographic binding between the leaf node representative of the candidate data item and the root node. At step 1401, the variable i is initialized to the number of the leaf node in the tree, n is initialized to the number of nodes in the tree, x is initialized to zero, R is initialized to the leaf node (i.e., $N_{0,i}$), and k is initialized to the number of supporting hashes provided. Note that n was checked along with the digital-signature on the root node. At step 1402, the device checks whether i is even and equal to $n-1$. If so, the device skips directly to step 1408, but otherwise the device proceeds to step 1403 and increments j . At step 1404, the device ensures that j (the number of supporting hashes used) never exceeds k (the total number of supporting hashes provided). At step 1405, the verifier determines whether i is even or odd. If i is odd, the verifier proceeds to step 1406 where the new value for R is found by concatenating the existing R with the next supporting hash (" \parallel " denotes concatenation) and hashing the result with a cryptographic hash function. If i is even, the verifier proceeds instead to step 1407, where the new value for R is found by

concatenating the existing next supporting hash with the existing
R (in the opposite order from step 1406) and hashing the result
with a cryptographic hash function. After step 1406 or 1407, the
verifier proceeds to step 1408 and divides i by 2 (rounding
5 upward), divides n by 2 (rounding upward), and increments x. At
step 1409 the verifier checks whether the main loop is complete,
returning to step 1402 if n has not yet reached one. If the loop
has finished, the verifier finally checks, at step 1411, whether
R corresponds to the expected value of the root node from the
confirmation. If R corresponds to the root node, the verifier
concludes at step 1412 that the binding is good. If R does not
correspond to the root node, the verifier concludes at step 1413
that the binding is bad.

FIG. 15 outlines the operation of a system which uses
the invention to determine whether certificates have been
revoked. The tree issuer 1501 constructs a list of revoked
certificates by obtaining CRLs or other revocation messages 1502.
The tree issuer then constructs an interval hash tree including
the digitally signed root node(s). The confirmation issuer 1503
20 obtains the tree's signed root from the tree issuer over a
communications channel. The confirmation issuer also obtains,
typically also from the tree issuer, the rest of the tree or the

leaf nodes needed to reconstruct the tree. The owner of a
certificate 1504 submits its certificate to the confirmation
issuer, which responds with a confirmation. The certificate
holder can then provide the confirmation along with its
5 certificate to certificate acceptors 1505, which each verify the
confirmation message to confirm that the certificate has not been
revoked. Participants in the protocol can verify the operation
of the tree issuer to detect attempts at fraud. In particular,
the tree should include all revoked certificates with no
10 unauthorized additions. For every certificate included in the
tree, the tree issuer should be able to provide a CRL or other
acceptable evidence of revocation. In some cases, such as if a
CA stops issuing CRLs, the tree issuer can optionally define
alternate mechanisms for adding entries to the list. In general,
the tree issuer can determine the criteria for including items in
the list, allowing the addition of new revocation mechanisms,
such as revocation by certificate holders. Even so, the
operation of the tree issuer is open to public scrutiny. In
particular, third parties can verify that the tree's leaf nodes
20 specify only properly-revoked certificates and that no revoked
certificates were omitted. The third party can also confirm that

the leaf nodes' ranges were constructed properly and detect any other abnormalities.

Those skilled in the art will appreciate that many simple variant forms of the invention are possible. For example, in another embodiment of the invention, the hash tree can be constructed using the sorted list of items (rather than ranges) as leaf nodes, in which case confirmations will consist of a pair of adjacent leaf nodes whose values span the candidate item, the intermediate nodes connecting the leaf nodes to the root, and the digitally-signed root. It is also possible to combine the functionality of the tree issuer and confirmation issuer within a single entity. In yet another embodiment of the invention, a trusted confirmation issuer can generate digitally-signed assurances as to the status of candidate items on the list, rather than including a chain of nodes from the interval hash tree as part of the confirmation. Finally it is also possible for the confirmation issuer to issue confirmations without receiving an explicit request.

20 Conclusions

Accordingly, the reader will see that this invention can be used to efficiently and securely demonstrate the presence

or absence of items on a list. Traditional solutions to this problem are inefficient in that an amount of data must be downloaded that is proportional to the number of items on the list. In contrast, the present invention provides tremendous savings for large lists. For example, by using binary interval hash trees, the amount of data required is proportional to the base-2 logarithm of the total list size. Construction of such hash trees is easy, requiring only $(2n-1)$ cryptographic hash operations, where n is the number of leaves in the tree.

Furthermore, construction of confirmation messages can be done efficiently and easily using insecure hardware since no private key operations are performed by the confirmation issuer.

Reliable and efficient verification of certificate revocation status is absolutely essential for secure World Wide Web transactions, Internet-based EDI, distributed design and manufacturing collaboration, financial transaction security, secure exchange of electronic mail, communication of medical records, and other applications requiring digital certificates. Using this invention, parties holding digital certificates can obtain tree leaves corresponding to their own certificates and supply the tree and corresponding signature information along with the certificate. This eliminates the need for certificate

recipients to independently download a CRL. Furthermore, the number of extra network connections involved in certificate revocation status is small, since a certificate holder can use a single nonrevocation proof for many transactions.

5 Although the description above contains many specificities, these should not be construed as limiting the scope of the invention but merely providing illustrations of some of the exemplary embodiments thereof. For example, the security system can be used with many sorts of data, including but not limited to lists of revoked certificates, revoked digital signatures on computer-executable code (such as executable code, object code, source code, interpreted code, etc.), lists of other types of revoked digital signatures of other types, lists of authorized users, lists of unauthorized users, lists of stolen credit card numbers, and lists of bad checks. In general, the invention is useful whenever it is necessary to determine in a secure manner whether or not a particular value is on a list. The system can be implemented using almost any computer technology, including almost any software language (Pascal, C, 20 C++, assembly language, FORTRAN, etc.) or integrated circuit manufacturing technology.

CLAIMS

What is claimed:

1 1. A method comprising the computer implemented steps of:
2 sorting a plurality of data items belonging to a superset of data items;
3 deriving a plurality of ranges using adjacent pairs of data items in said sorted
4 plurality of data items as endpoints such that all data items in said plurality
5 of the data items are at endpoints of said plurality of ranges and such that
6 all other data items in said superset fall in-between the endpoints of said
7 plurality of ranges;
8 generating a hash tree having leaf nodes that represent the plurality of ranges;
9 digitally signing a root node of the tree; and
10 electronically transmitting said digitally signed root node and parts of said tree
11 onto a network for use in cryptographically demonstrating whether a given
12 data item is one of said plurality of data items.

1 2. The method of claim 1, wherein said step of generating said tree includes the step
2 of:
3 forming leaf nodes from endpoints of said plurality of ranges.

1 3. The method of claim 2, wherein said step of generating said tree includes the step
2 of:
3 forming an adjacent pair of leaf nodes from different endpoints of one of said
4 plurality of ranges.

1 4. The method of claim 1, wherein said step of generating said tree includes the step
2 of:

003388.P001C "543210"

3 forming each of a plurality of the leaf nodes from the endpoints of a different one
4 of said plurality of ranges.

1 5. The method of claim 1, wherein said plurality of data items identify digital
2 certificates sharing an attribute.

1 6. The method of claim 5, wherein said attribute is that the digital certificates are
2 revoked.

1 7. The method of claim 1, wherein said plurality of data items identify digital
2 signatures.

1 8. The method of claim 1, wherein said plurality of data items identify digital
2 signatures on binary code.

1 9. The method of claim 1, wherein said plurality of data items identify revoked
2 credit cards.

1 10. A method comprising the computer implemented steps of:
2 receiving a request message requesting whether a first data item is one of a
3 plurality of data items belonging to a superset of data items;
4 selecting a range that is derived from the pair of data items in said plurality of data
5 items that defines the smallest range that includes said first data item,
6 wherein the first data item is not one of the plurality of data items if the
7 first data item is in-between the endpoints of the selected range, and
8 wherein the first data item is one of said plurality of data items if said first
9 data item is on one of the endpoints of the selected range;

determining, for a tree having leaf nodes that represent ranges derived from adjacent pairs of said plurality of data items in a sorted list of said plurality of data items, a path through said tree from said selected range to a first of a set of root nodes; and

generating a response message that includes,

data identifying said selected range in said response message,

a set of nodes in said tree such that each node in said set of nodes and at least a previously identified node on said path can be combined to identify a previously unidentified node on said path, until said first root node is identified,

the set of nodes and excluding at least certain nodes on the path from the response message, and

a digitally signed representation of the first root node in said response message; and

electronically transmitting the response message onto a network.

11. The method of claim 10, wherein the endpoints of the selected range are independently hashed and then hashed together to generate a node in the tree.

12. The method of claim 10, wherein each leaf node specifies one of one range, an endpoint of one range, a hashed range, and a hashed endpoint of one range.

13. The method of claim 10, wherein said plurality of data items identify digital certificates sharing an attribute.

14. The method of claim 13, wherein said attribute is that the digital certificates are revoked.

1 15. The method of claim 10, wherein said plurality of data items identify digital
2 signatures.

1 16. The method of claim 10, wherein said plurality of data items identify digital
2 signatures on binary code.

1 17. The method of claim 10, wherein said plurality of data items identify revoked
2 credit cards.

1 18. A method comprising the computer implemented steps of:
2 electronically transmitting a request message as to whether a first data item is one
3 of a plurality of data items belonging to a superset of data items;
4 receiving a response message that includes,
5 a digitally signed representation submitted to be a root node of a tree
6 having leaf nodes that represent ranges, the ranges derived from
7 adjacent pairs of data items in a sorted list of said plurality of data
8 items,
9 data identifying the range that includes said first data item, and
10 a set of nodes in said tree sufficient to generate said root node starting
11 from said range; and
12 determining if said first data item is one of said plurality of data items based on
13 said range, wherein the first data item is not one of the plurality of data
14 items if the first data item is in-between the endpoints of the selected
15 range, and wherein the first data item is one of said plurality of data items
16 if said first data item is on one of the endpoints of the selected range; and
17 generating said root node using said range and the set of nodes; and

003388.P001C "349T50"

18 determining if said digitally signed representation matches said root node.

1 19. The method of claim 18, wherein said set of nodes includes nodes that together
2 with a node on a path from the range to the root node can be used to generate a next node
3 on the path, but excluding at least some nodes that are on the path.

1 20. The method of claim 18, wherein said step of receiving the response message
2 includes the step of:
3 generating a node in said tree from the range that includes the first data item and
4 another range identified in said response message.

1 21. The method of claim 18, wherein each node in said set of nodes and at least a
2 previously identified node on a path from the range to the root node can be combined to
3 identify a previously unidentified node on said path, until said root node is identified.

1 22. The method of claim 18, wherein each leaf node specifies one of one range, an
2 endpoint of one range, a hashed range, and a hashed endpoint of one range.

1 23. The method of claim 18, wherein said plurality of data items identify digital
2 certificates sharing a attribute.

1 24. The method of claim 23, wherein said attribute is that the digital certificates are
2 revoked.

1 25. The method of claim 18, wherein said plurality of data items identify digital
2 signatures.

ABSTRACT

Methods and apparatuses for providing cryptographic assurance based on ranges as to whether a particular data item is on a list. According to one computer-implemented method, the items on the list are sorted and ranges are derived from adjacent pairs of data items on the list. Next, cryptographically manipulated data is generated from the plurality of ranges. At least parts of the cryptographically manipulated data is transmitted onto a network for use in cryptographically demonstrating whether any given data item is on the list. According to another computer-implemented method, a request message is received requesting whether a given data item is on a list of data items. In response, a range is selected that is derived from the pair of data items on the list that define the smallest range that includes the given data item. A response message is transmitted that cryptographically demonstrates whether the first data item is on the list using cryptographically manipulated data derived from the range. According to another computer-implemented method, a request message requesting an indication as to whether a first data item is on a list of data items is transmitted. In response, a message is received that cryptographically demonstrates whether the first data item is on the list, where the response message identifies a range that is derived from the pair of data items on the list that defines the smallest range that includes the first data item.

FIG. 1 (Prior Art)

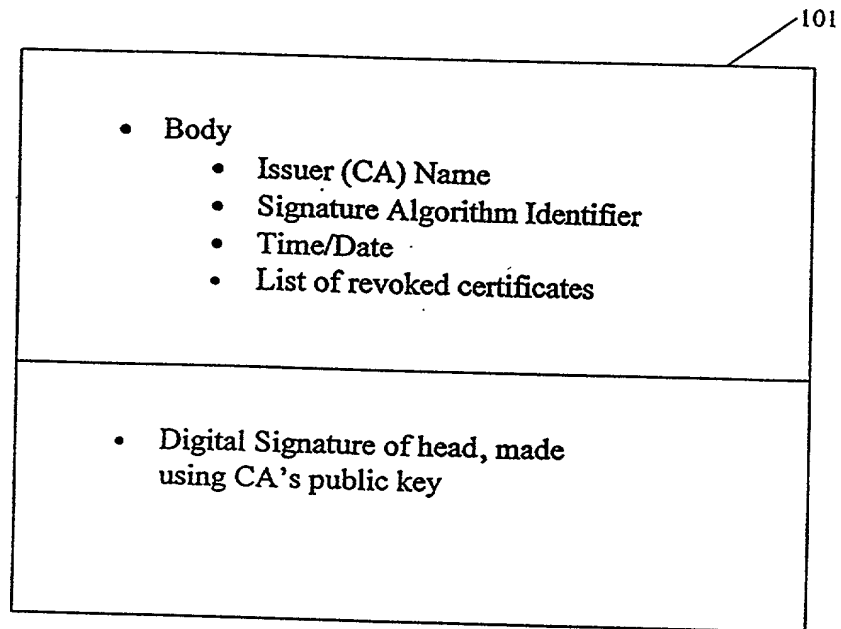
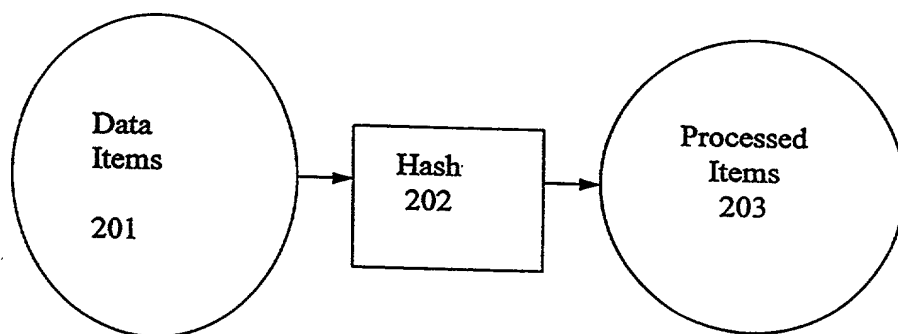
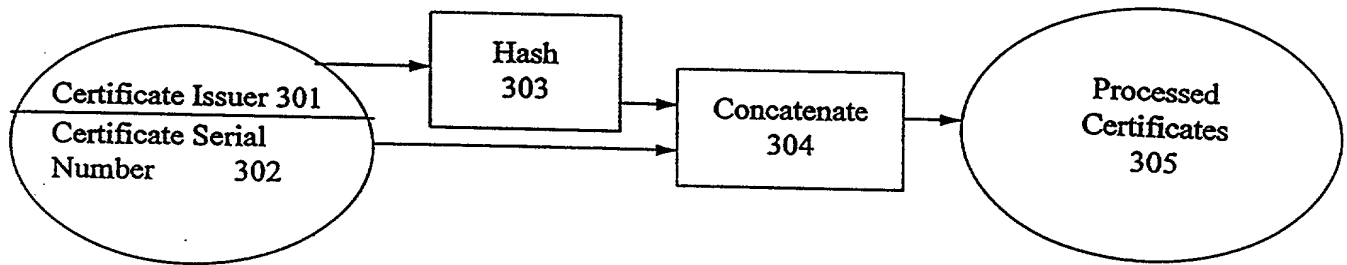


FIG. 2



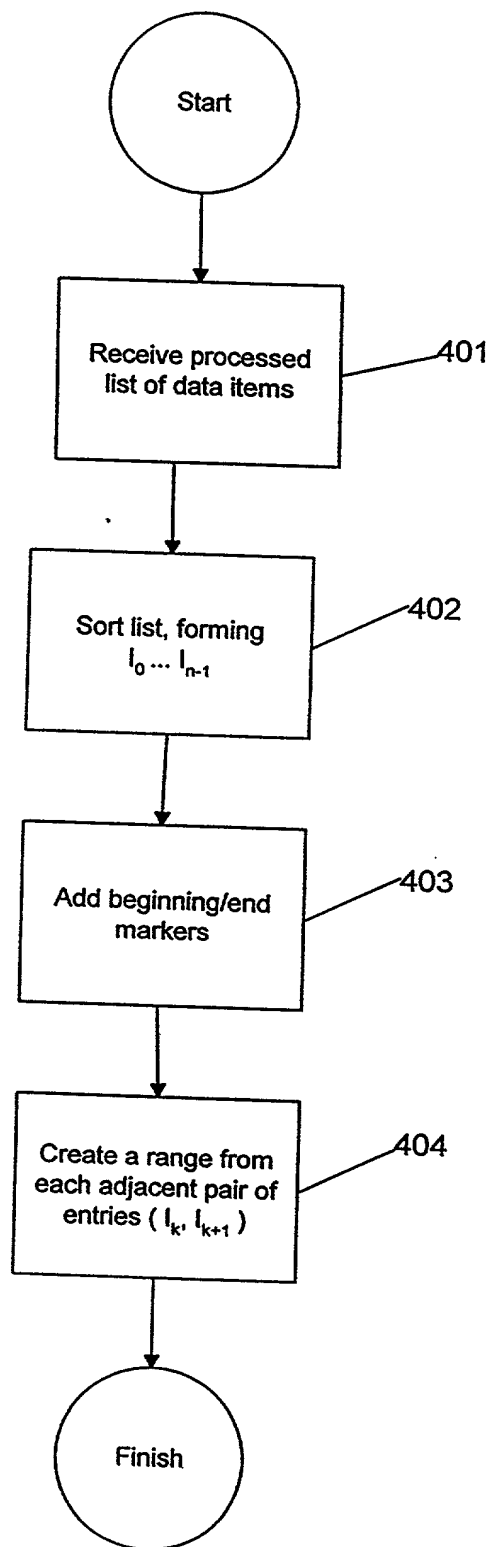
09511645, 022300

FIG. 3



09311645-022200

FIG. 4



002220" 5497T.550

FIG. 5.

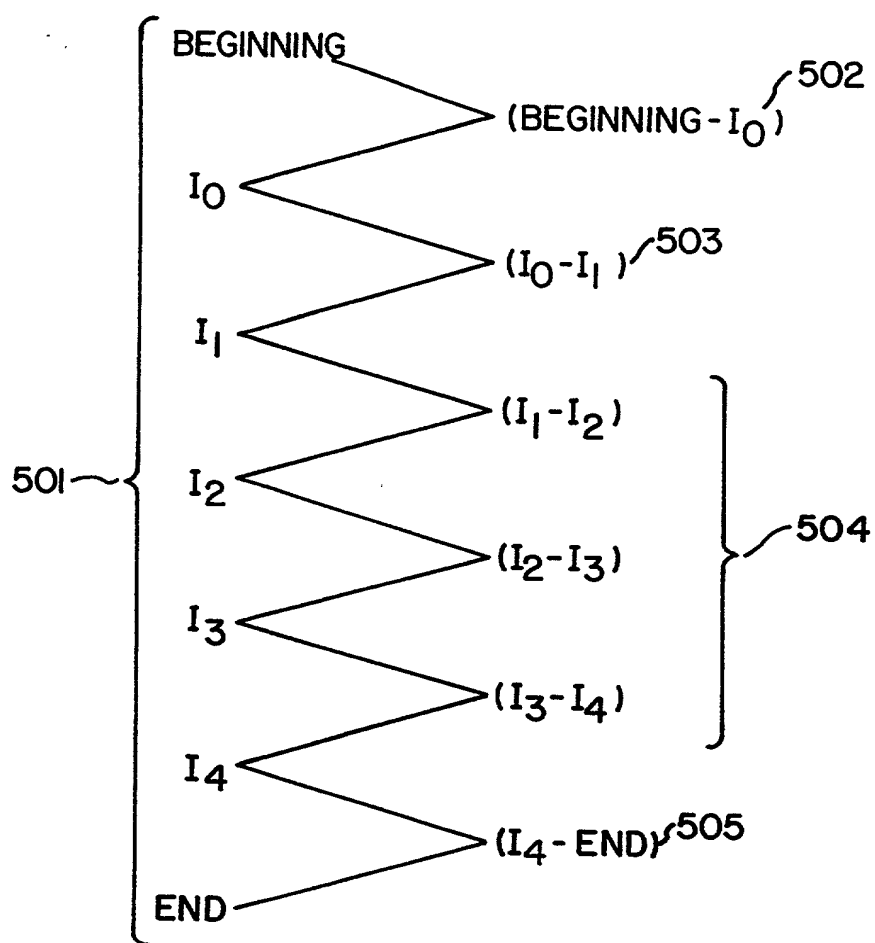


FIG. 6.

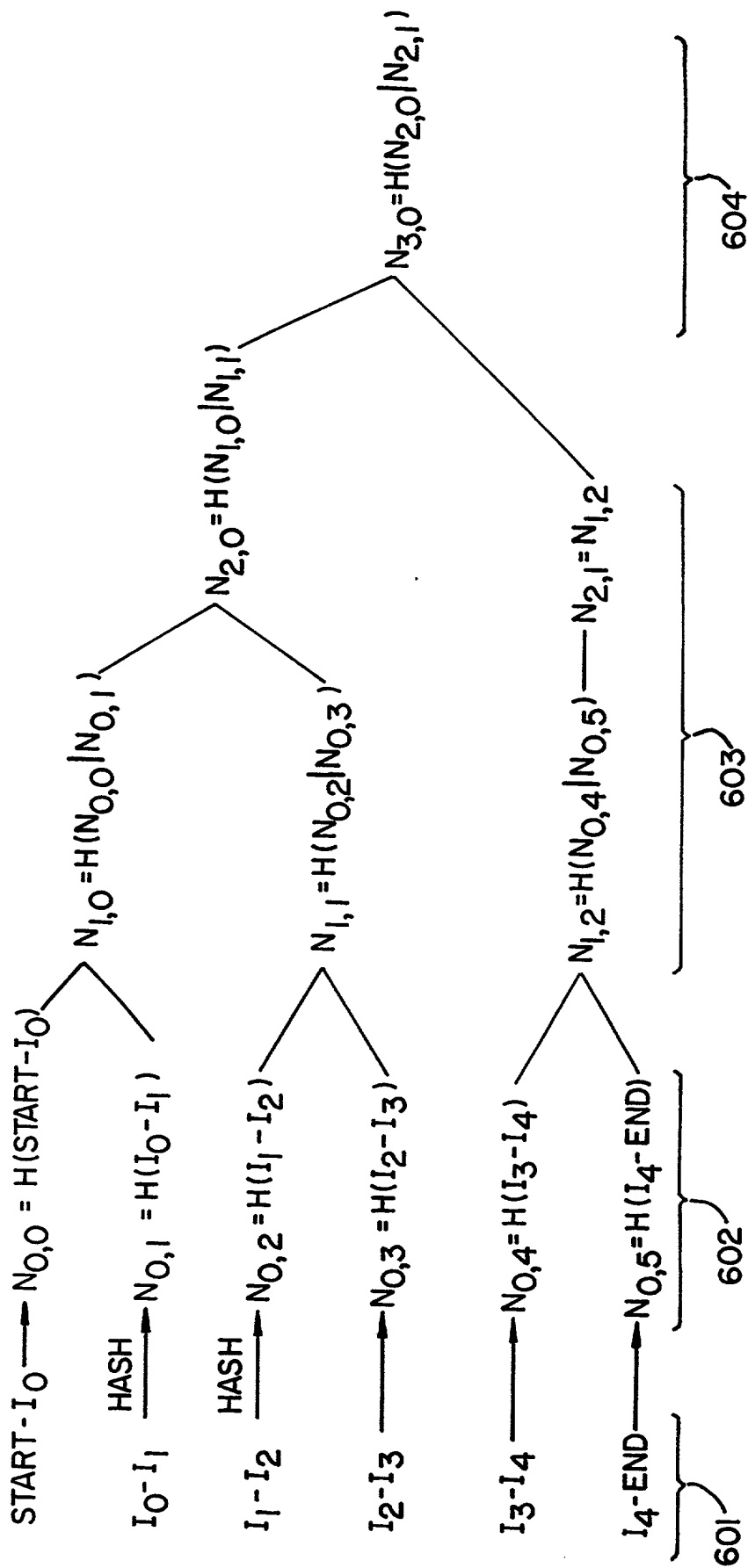
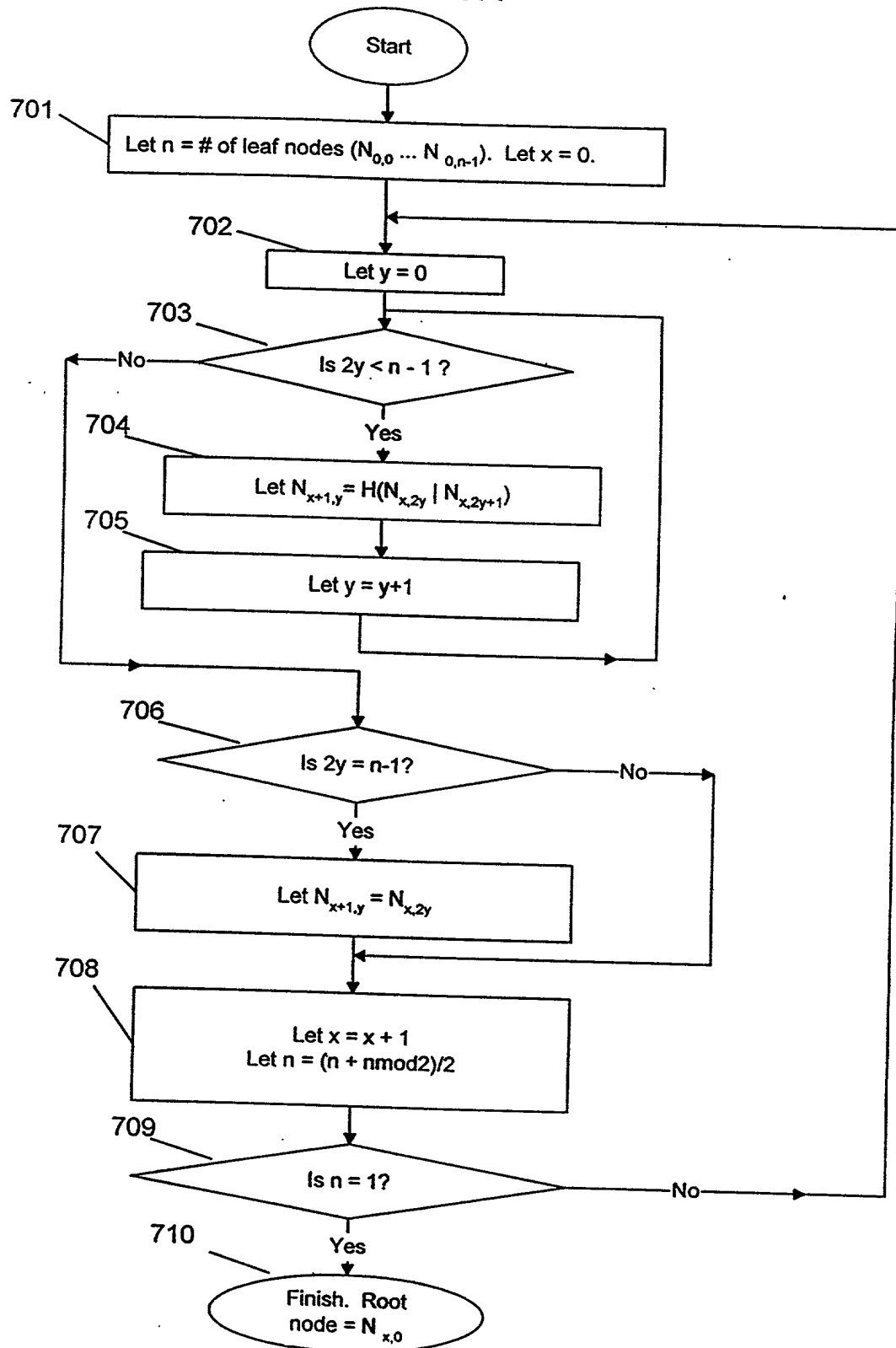


FIG. 7



00220" 5491" 560

FIG. 8.

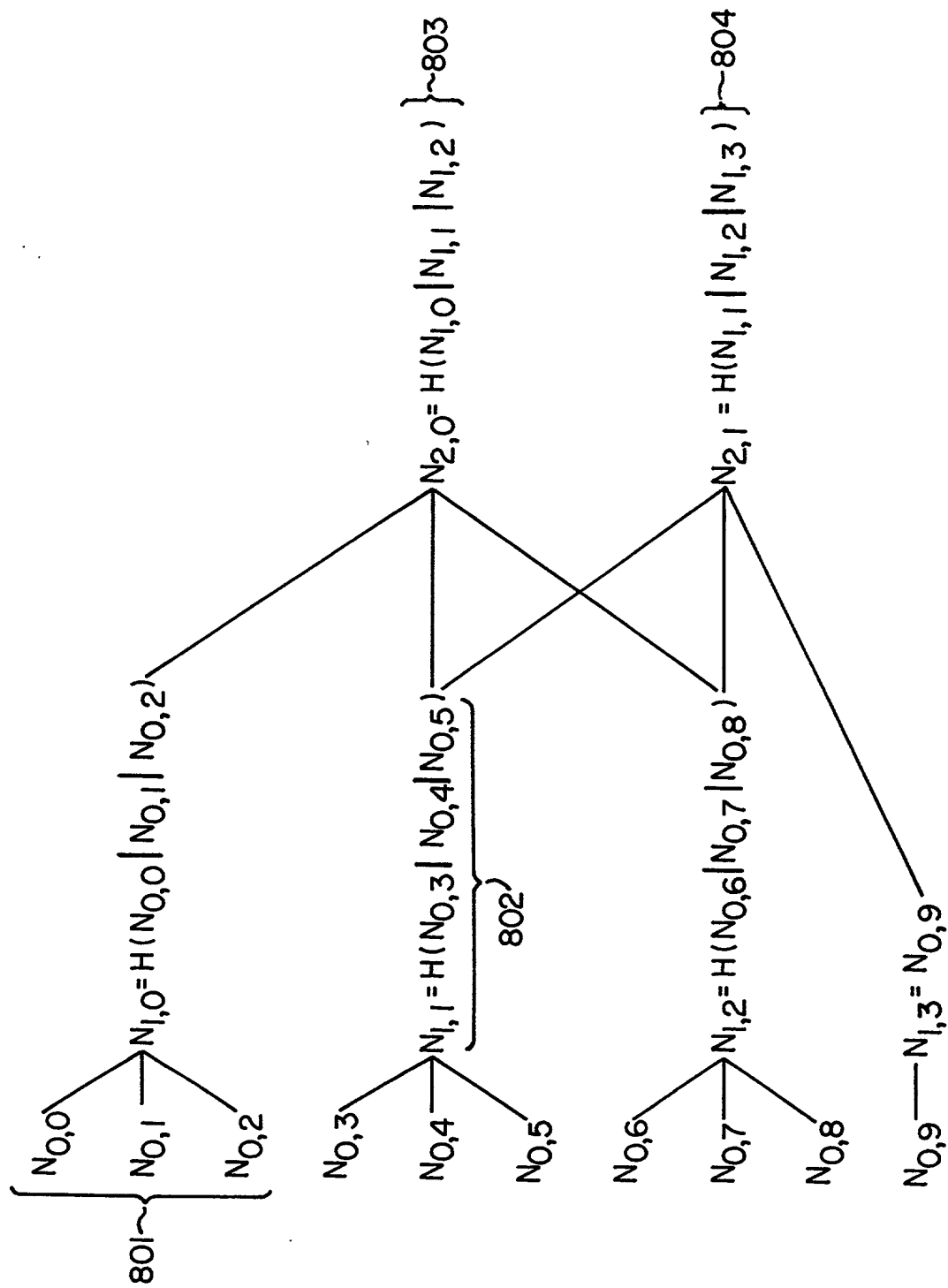
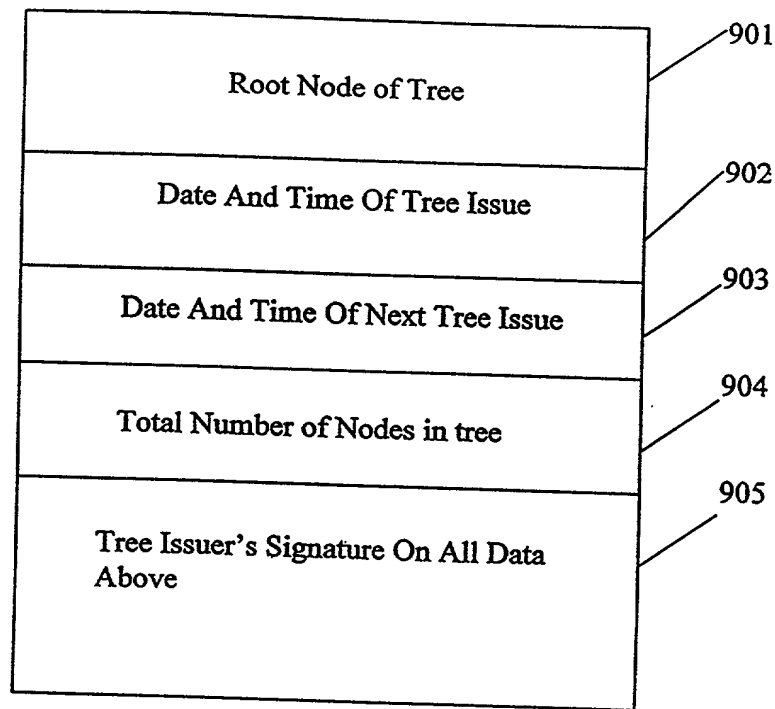


FIG 9



09511645 022200

FIG. 10.

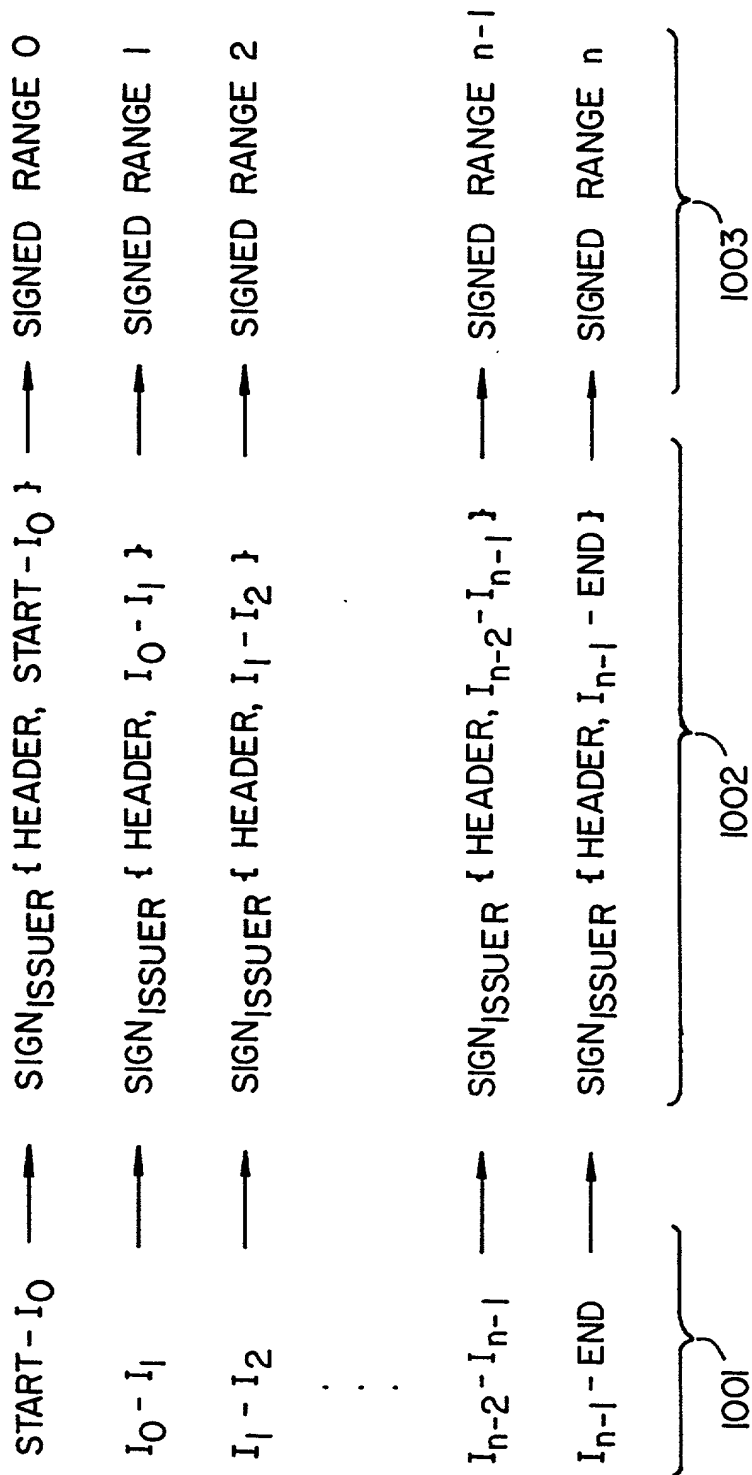
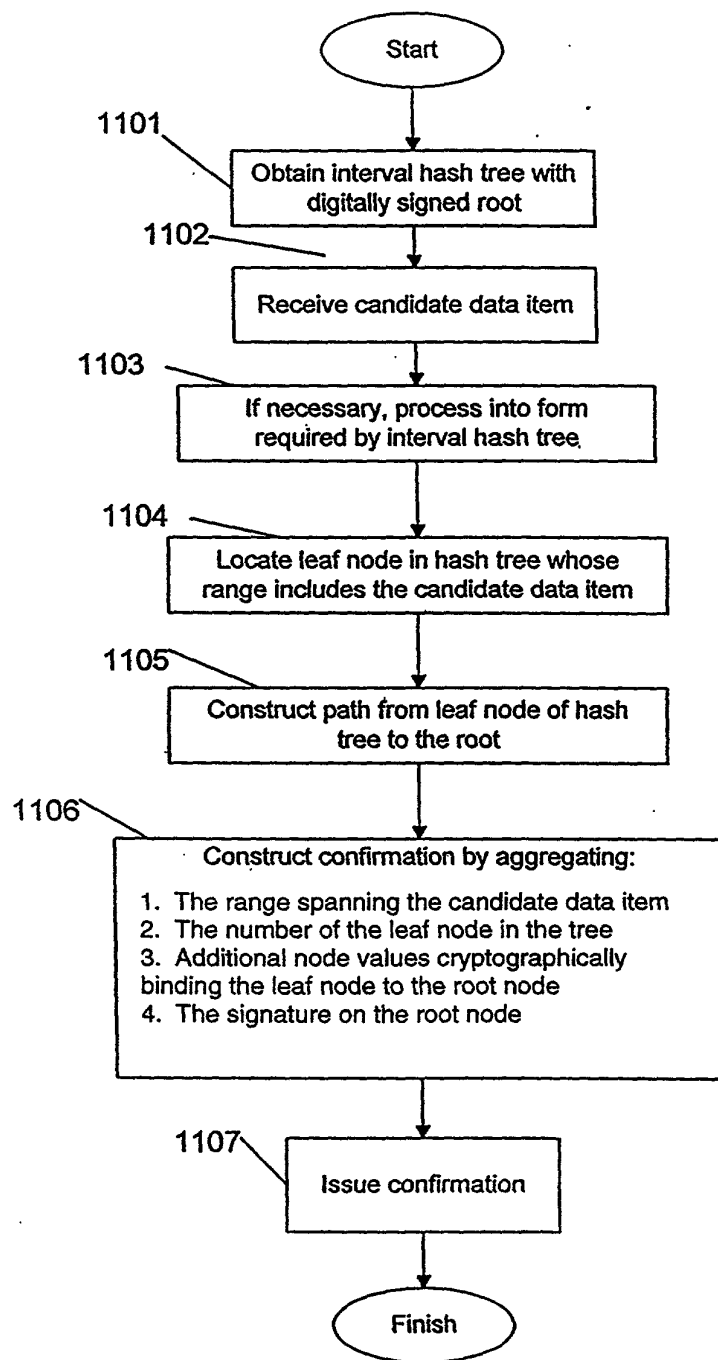
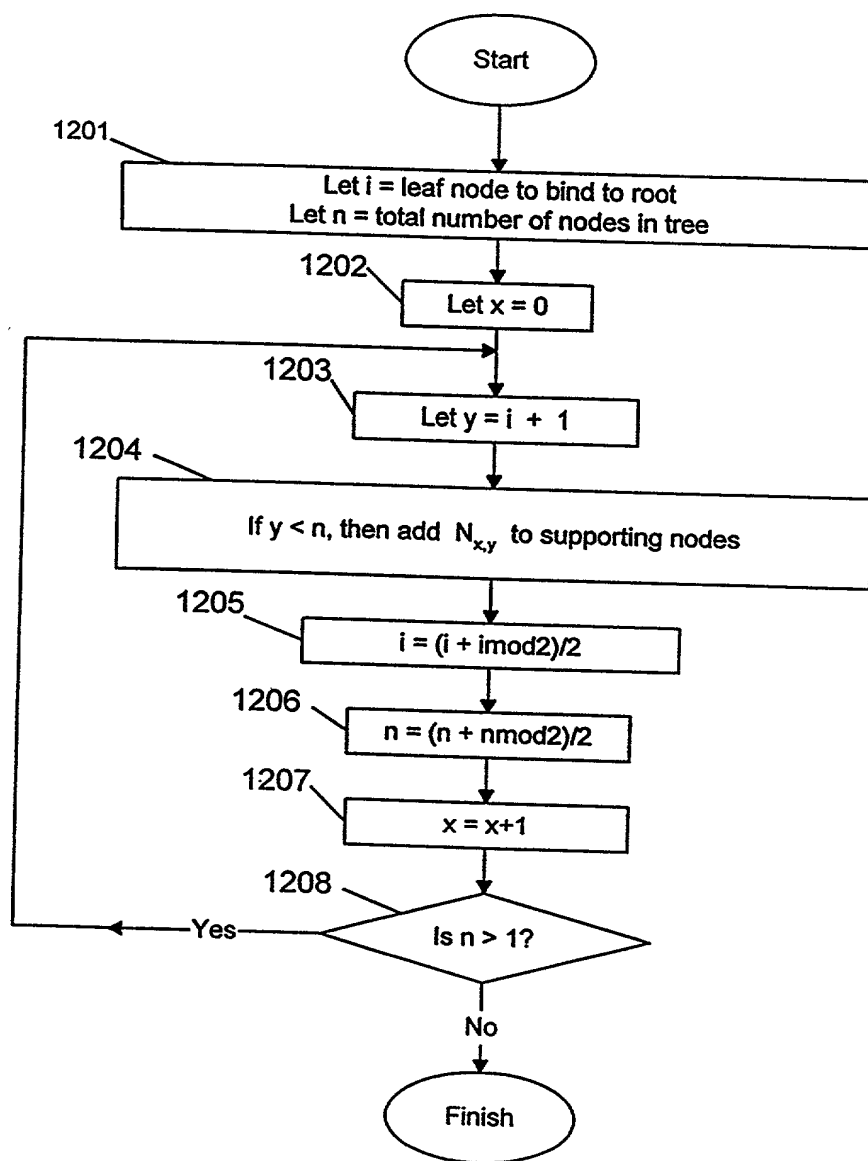


FIG. 11



0934345-02200

FIG. 12



BOOK REVIEW

FIG. 13

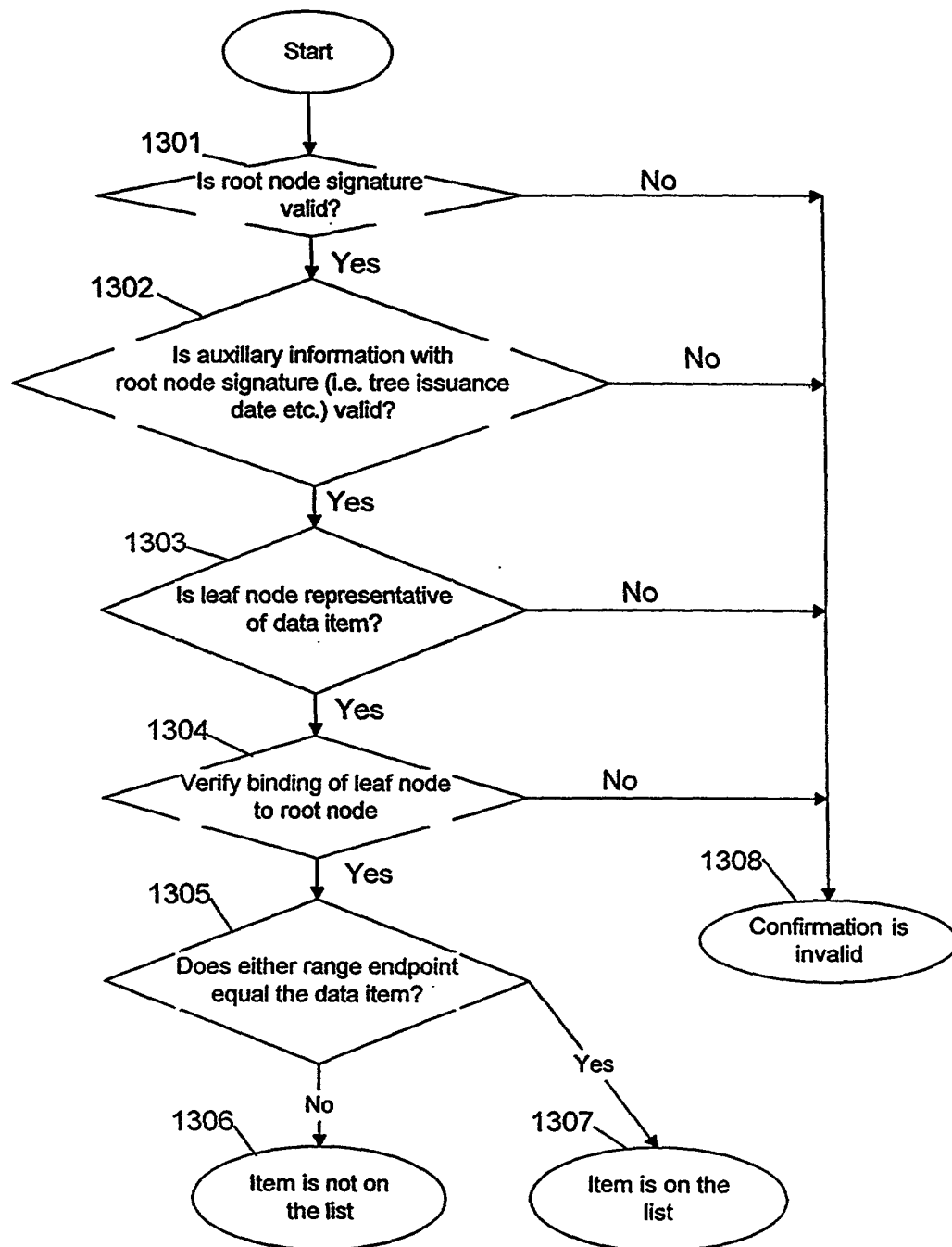
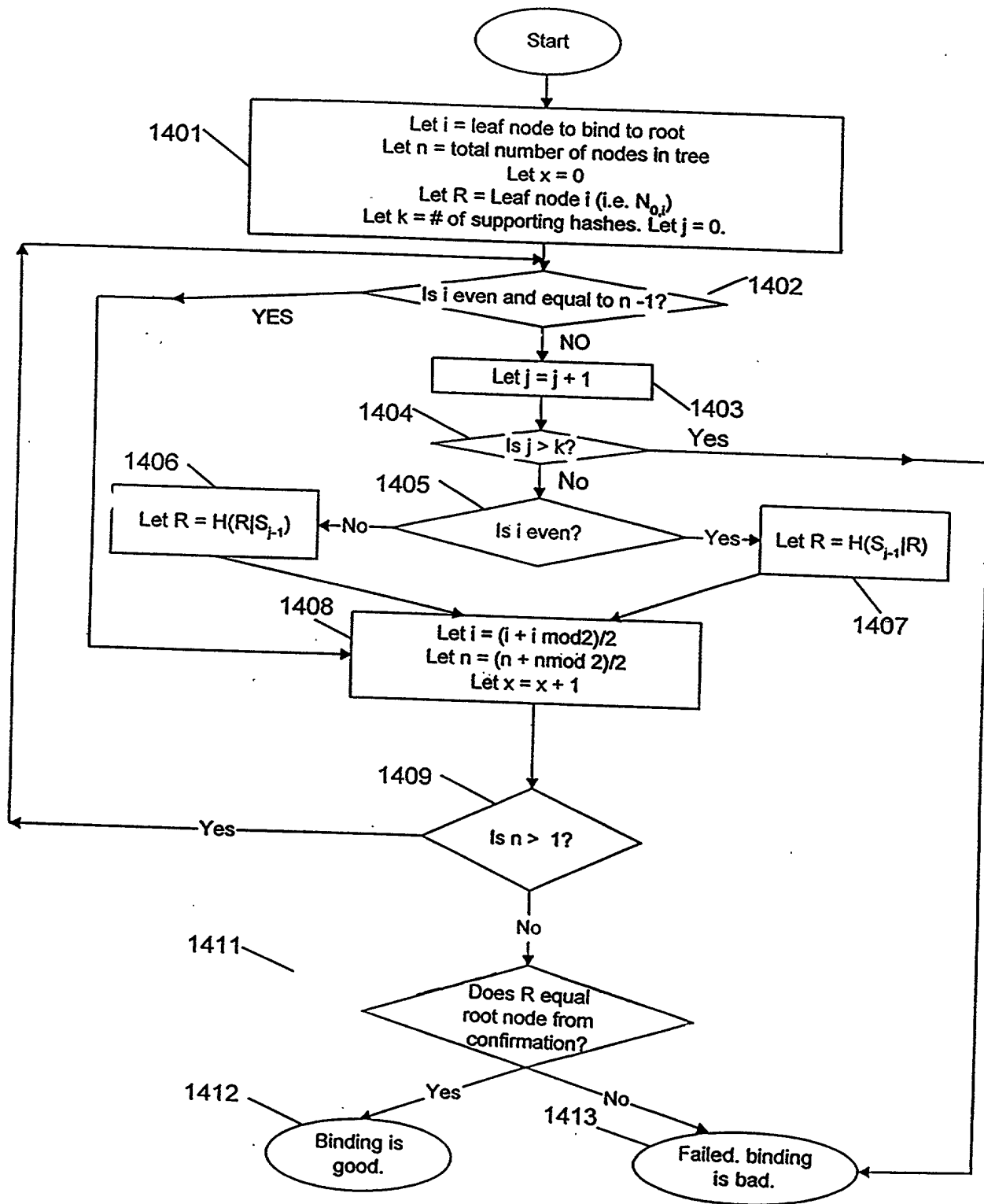
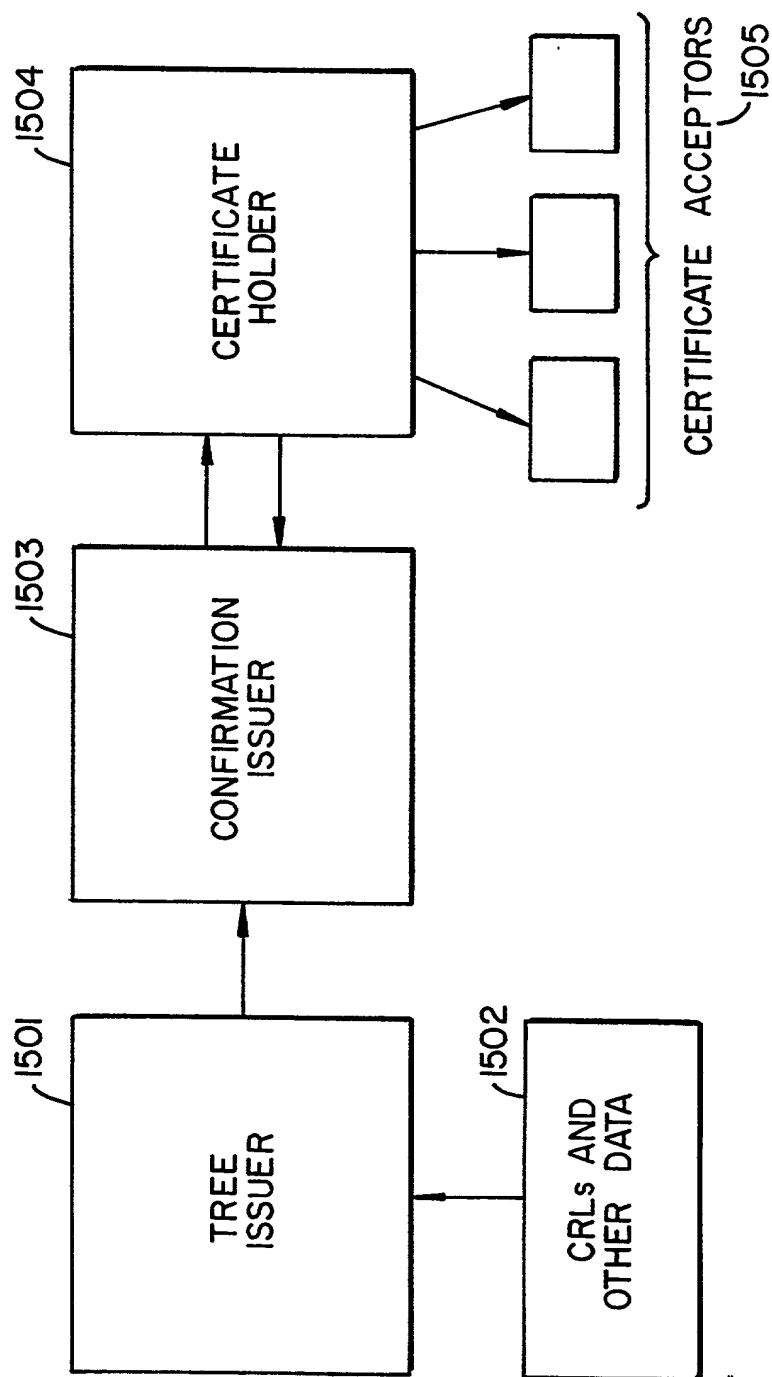


FIG. 14



00220" 549TTS60

FIG. 15.



DECLARATION FOR PATENT APPLICATION

As a below named inventor(s), I/we hereby declare that:

Our residences and citizenship are as stated below next to our names.

I/We believe I/we am/are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled **APPARATUS AND METHOD FOR DEMONSTRATING AND CONFIRMING THE STATUS OF DIGITAL CERTIFICATES AND OTHER DATA** the specification of which

 X is attached hereto.

 was filed on [date] as
Application Serial Number []
and was amended on [date].

I/We hereby state that we have reviewed and understand the contents of the above-identified specification, including the claims as amended by any amendment referred to above.

I/We acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I/We hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate, listed below and so identified, and we have also identified below any foreign application for patent or inventor's certificate on this invention filed by me or my legal representatives or assigns and having a filing date before that of the application on which priority is claimed.

Number	Country	Day/Month/ Year Filed	Priority Claimed - Yes or No

I/We hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, we acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

Application Serial No.	Filing Date	Status

I/We hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: **Paul Carl Kocher**

Inventor's signature:

Paul C Kocher

Date: 13 MAY, 1996

Residence:

2103 Menalto Avenue
Menlo Park, CA 94025

Citizenship:

USA